Politecnico di Torino

Digital systems electronics
A.A. 2018/2019

Prof. G. Masera

Lab 03

2 Apr 2019

Berchialla Luca	236032
Laurasi Gjergji	238259
Mattei Andrea	233755
Lombardo Domenico Maria	233959

1 4-bit Sequential RCA

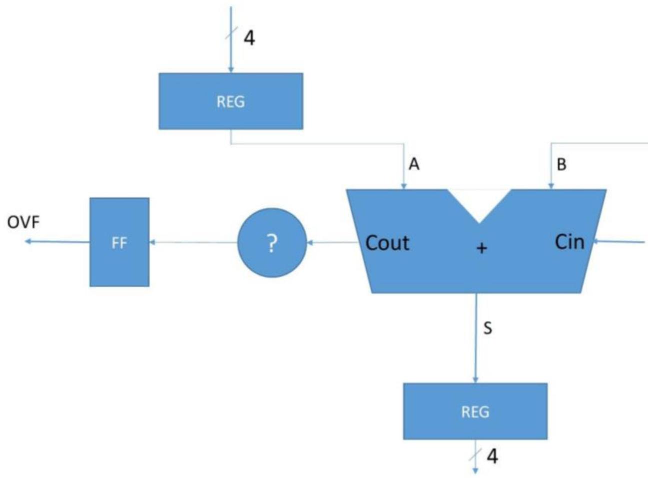
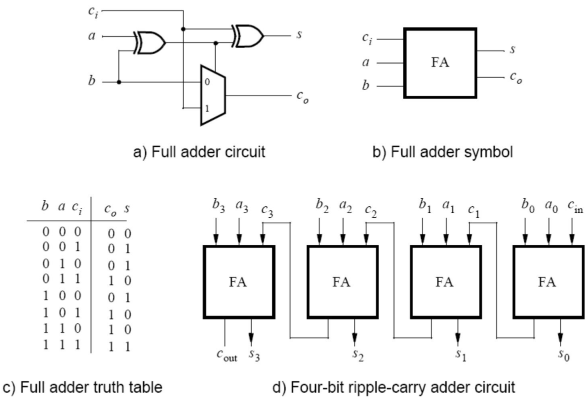


Figure 1: Top level entity

Here we had to implement a 4-bit Sequential Ripple Carry Adder. To build the circuit requested in $figure1$ several sub-circuits have been implemented.
As first point a Full Adder was implemented as shown in $figure\ 2a$. The 4-bit adder was built using four full adders in a ripple carry architecture. Its overflow signal is generated by a xor gate whose inputs are the $carry_{out}$ signals of the last two full adders.



a) Full adder circuit

b) Full adder symbol

b	a	c_i	c_o	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

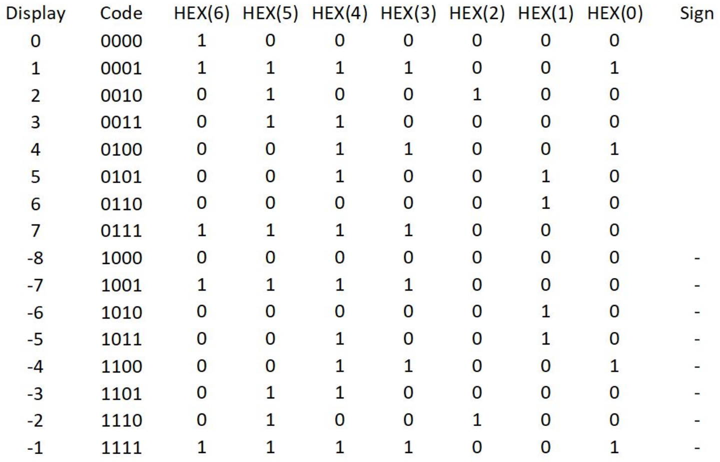
c) Full adder truth table

d) Four-bit ripple-carry adder circuit

Figure 2: Full Adder

The Register and the Flip-Flop were implemented using the code provided in the instructions.

A new 7-segments display decoder was needed in order to display properly 2's complement numbers. It uses two 7-segments displays to show respectively the sign and the magnitude of the number. The implementation was done by means of the $when - else$ statement following the truth table in $figure$ 3.



Display	Code	HEX(6)	HEX(5)	HEX(4)	HEX(3)	HEX(2)	HEX(1)	HEX(0)	Sign
0	0000	1	0	0	0	0	0	0	
1	0001	1	1	1	1	0	0	1	
2	0010	0	1	0	0	1	0	0	
3	0011	0	1	1	0	0	0	0	
4	0100	0	0	1	1	0	0	1	
5	0101	0	0	1	0	0	1	0	
6	0110	0	0	0	0	0	1	0	
7	0111	1	1	1	1	0	0	0	
-8	1000	0	0	0	0	0	0	0	-
-7	1001	1	1	1	1	0	0	0	-
-6	1010	0	0	0	0	0	1	0	-
-5	1011	0	0	1	0	0	1	0	-
-4	1100	0	0	1	1	0	0	1	-
-3	1101	0	1	1	0	0	0	0	-
-2	1110	0	1	0	0	1	0	0	-
-1	1111	1	1	1	1	0	0	1	-

Figure 3: 2's complement 7-segments decoder

The top level entity that implements the circuit is in the file $lab3\_es1.vhd$ where all the components needed are instantiated and connected together.
As required, the the inputs A and B have been assigned to SW_{3-0} and SW_{7-4} respectively, KEY_0 to the negated asynchronous reset input and KEY_1 to the clock. The magnitude and the sign of A are displayed in $HEX4$ and $HEX5$ respectively. The same is done for B and S in $HEX2, HEX3$ and $HEX0, HEX1$ respectively. The adder's overflow is shown by means of the red $LEDR_9$.

1.1 Testbench

To validate the correct behavior of the circuit a testbench was created. The *clock* signal was created using a process and its period has been fixed to $10ps$ ps

2 4-bit Sequential Adder/Subtractor

3 3

punto 3

4 4

punto 4